

自由存储区和堆

自由存储区和堆的区别是一个老生常谈的问题了，就像面试题 `new` 和 `malloc` 的区别一样。事实上，网上多说人认为，自由存储区与堆的划分标准是申请和释放内存是使用的 `new/delete` 还是 `malloc/free`。C++ 标准并没有给出 `new/delete` 应该如何实现，但很多编译器的 `new/delete` 都是以 `malloc/free` 为基础来实现的。从技术上来说，堆（heap）是 C 语言和操作系统的术语，堆是操作系统所维护的一块特殊内存，它提供了动态分配的功能，使用 `malloc()`、`free()` 来申请/释放内存。而自由存储是 C++ 中通过 `new` 和 `delete` 动态分配和释放对象的抽象概念。基本上，所有的 C++ 编译器默认使用堆来实现自由存储。也就是说，默认的全局运算符 `new` 和 `delete` 也许会使用 `malloc` 和 `free` 的方式申请和释放存储空间，也就是说自由存储区就位于堆上。但程序员也可以通过重载操作符，改用其他内存来实现自由存储，例如全局变量做的对象池，这时自由存储区就不位于堆上了。

因此，我认为自由存储区和堆有如下的区别：**堆是操作系统维护的一块内存，是一个物理概念，而自由存储是 C++ 中通过 `new` 与 `delete` 动态分配和释放的对象的存储区，是一个逻辑概念。**

PS：C++ 的内存布局：内存区分为 5 个区，分别是堆、栈、自由存储区、全局/静态存储区、常量存储区